Data Integration and Large-Scale Analysis

Winter Semester 2025/26

Course Project: Entity Matching Pipeline

Deadline: January 10th 2026, **Score to pass:** 51 points

Task 01: Entity Matching Pipeline (40 Points)

You are given two bibliographic datasets (*DBLP1.csv, Scholar.csv*) and a ground-truth mapping (*DBLP-Scholar_perfectMapping.csv*). The goal is to build an end-to-end pipeline to identify matching records referring to the same publication.

1.1 Data Preparation (10 pts)

- Clean and normalize text; tokenize title, extract author last names, parse year
- Deduplicate using Jaccard ≥ 0.95 or relaxed rule

Report: number of input records, duplicate groups, final unique records

1.2 Blocking (10 pts)

- Apply ≥2 blocking rules (e.g., title prefix, author overlap…)
- Generate union of matches; report candidate pairs count, and % gold retained

1.3 Similarity Scoring (10 pts)

- Compute cosine similarity (TF-IDF)
- Also explore Jaccard, Dice, Levenshtein, or field-specific scores

Report: number of pairs with cosine ≥ 0.95 + comparison summary

1.4 Matching & Evaluation (10 pts)

- Evaluate against perfect mapping

Report: P/R/F1 at thresholds: 0.70, 0.80, 0.90, 0.95

Task 02: Feature Vector and ML Model (50 Points)

Train a Machine Learning model to classify whether a candidate pair refers to the same entity.

2.1 Create a Training Dataset (10 pts)

Start from the output pairs generated in Task 1 (e.g., pairs_scored.csv, which contains all candidate pairs and their similarity scores).

- Assign label = 1 if the (DBLP, Scholar) pair exists in the gold mapping
- Label = 0 otherwise

Hint: You could use *DBLP-Scholar_perfectMapping.csv* to identify true matches.

2.2 Feature Engineering (10 pts)

- Compute at least 6 features using existing similarity scores (title_cos, title_jacc, auth_olap, year_ok, venue_exact)

Hint: For better training data introduce additional features such as:

- Title length difference
- Jaccard/Dice similarity (authors or venue)
- Shared token count or character overlap
- Levenshtein distance
- Cosine similarity on other fields

2.3 Preprocess Training Data (10 pts)

- Handle missing values and duplicates
- Normalize or scale numeric features
- Address class imbalance using one of:
  - Undersampling
  - Class weights or

- SMOTE

2.4 Model Training (10 pts)

- Train both **Random Forest** and **SVM**
- Use **3-fold cross-validation**
- Report **Precision, Recall, and F1-score**

2.5 Evaluation (10 pts)

- Beat the baselines:
  - **SVM: F1 ≈ 0.76**
  - **RF: F1 ≈ 0.79**
- Discuss model performance and most useful features

Task 03: Reporting & Reproducibility (10 Points)

- Submit a precise and clear **report (PDF or Markdown)** that includes:
  - Overview of Tasks 1, 2, 3
  - Tables with summaries:
    - Blocking results
    - Similarity scores
    - ML evaluation (P/R/F1)
  - Feature engineering descriptions
- In addition to the previous report, your submission **must** include:
  - All Python/Jupyter code
  - requirements.txt
  - README.md with run instructions
- **All previous files should be compressed into a ZIP file.**